

The Chess - Java OOP Project

Oleh: Hansel Kristanzen Siswanto - 2802495505

Deskripsi Singkat

The Chess adalah aplikasi permainan catur berbasis GUI (Graphical User Interface) yang dibangun menggunakan bahasa pemrograman Java. Proyek ini dibuat untuk memenuhi tugas akhir mata kuliah **Pemrograman Berorientasi Objek (OOP)**.

Aplikasi ini tidak hanya menyajikan permainan catur standar, tetapi juga dilengkapi dengan fitur **Blitz Mode (Timer)**, pencatatan riwayat pertandingan ke dalam file **CSV**, serta validasi langkah bidak yang akurat.

Fitur Utama (4+ Fitur)

Sesuai kriteria penilaian, aplikasi ini memiliki lebih dari 4 fitur utama:

1. **Dua Mode Permainan:**
 - o **Unlimited Mode:** Bermain santai tanpa batasan waktu.
 - o **Blitz Mode:** Bermain kompetitif dengan timer mundur (10 menit per pemain).
2. **Sistem Validasi Gerak (Rule Engine):** Setiap bidak (Rook, Knight, Bishop, Queen, King, Pawn) memiliki logika validasi langkahnya sendiri sesuai aturan catur internasional.
3. **Riwayat Pertandingan (Match History):**
 - o Mencatat setiap langkah (Waktu, Giliran, Bidak, Posisi Awal, Posisi Akhir) ke dalam memori.
 - o **File Handling:** Menyimpan riwayat permanen ke dalam file eksternal chess_log.csv.
 - o Fitur untuk melihat kembali riwayat pertandingan sebelumnya melalui menu GUI.
4. **Fitur Surrender & Reset:** Pemain dapat menyerah di tengah permainan atau mereset papan untuk memulai ulang tanpa menutup aplikasi.
5. **Validasi Input & Exception Handling:** Mencegah input nama kosong dan menangani error saat membaca/menulis file.

Struktur Program & Daftar Class (Minimal 10 Class)

Proyek ini dirancang secara modular dengan total **14 Class** (termasuk Inner Classes), jauh melampaui batas minimal 10 class yang disyaratkan.

1. Abstract & Parent Classes

1. ChessGameProject (Main Class): Entry point aplikasi.
2. Piece (Abstract Class): Template dasar untuk semua bidak catur.

2. Concrete Classes (Inheritance - Bidak Catur)

3. King: Logika gerak Raja.
4. Queen: Logika gerak Ratu.
5. Rook: Logika gerak Benteng.
6. Bishop: Logika gerak Bishop.
7. Knight: Logika gerak Kuda.
8. Pawn: Logika gerak Pion.

3. Logic & Helper Classes

9. MoveValidator: Kelas utilitas statis untuk mengecek apakah jalur bidak terhalang atau tidak.
10. FileManager: Menangani operasi *File Input/Output* (Membaca dan menulis chess_log.csv).

4. GUI & Interface Classes

11. MainMenu (extends JFrame): Tampilan awal menu permainan.
12. ChessBoard (extends JFrame): Tampilan papan catur dan logika permainan inti.
13. HistoryViewer (extends JFrame): Jendela untuk melihat tabel riwayat CSV.
14. Cell (extends JButton): Representasi visual kotak pada papan catur.

Penerapan Konsep OOP

1. Encapsulation

Variabel data dibuat private dan diakses melalui method public (Getter).

- **Contoh Code:** Di class Piece:

```
private boolean isWhite;  
private String name;  
public boolean isWhite() { return isWhite; }
```

2. Inheritance

Menggunakan keyword extends untuk mewariskan sifat dari *Parent Class* ke *Child Class*.

- **Contoh:** Semua bidak (e.g., Rook, Knight) adalah turunan dari class Piece.
- **Contoh GUI:** ChessBoard dan MainMenu adalah turunan dari JFrame.

3. Polymorphism

Menggunakan @Override untuk mengubah perilaku method parent sesuai kebutuhan spesifik child class.

- **Contoh:** Setiap bidak memiliki implementasi method isValidMove() dan getSymbol() yang berbeda-beda meskipun nama methodnya sama.

```
// Di class Rook  
@Override  
public boolean isValidMove(...) { ...logika benteng... }
```

```
// Di class Knight  
@Override  
public boolean isValidMove(...) { ...logika kuda... }
```

4. Abstraction

Menyembunyikan detail implementasi yang kompleks menggunakan abstract class dan abstract method.

- **Contoh:** Class Piece tidak bisa diinstansiasi langsung dan memaksa turunannya untuk mengimplementasikan method abstrak:

```
public abstract boolean isValidMove(int startX, int startY, int endX, int endY, Piece[][]  
board);
```

File Handling & Koleksi Data

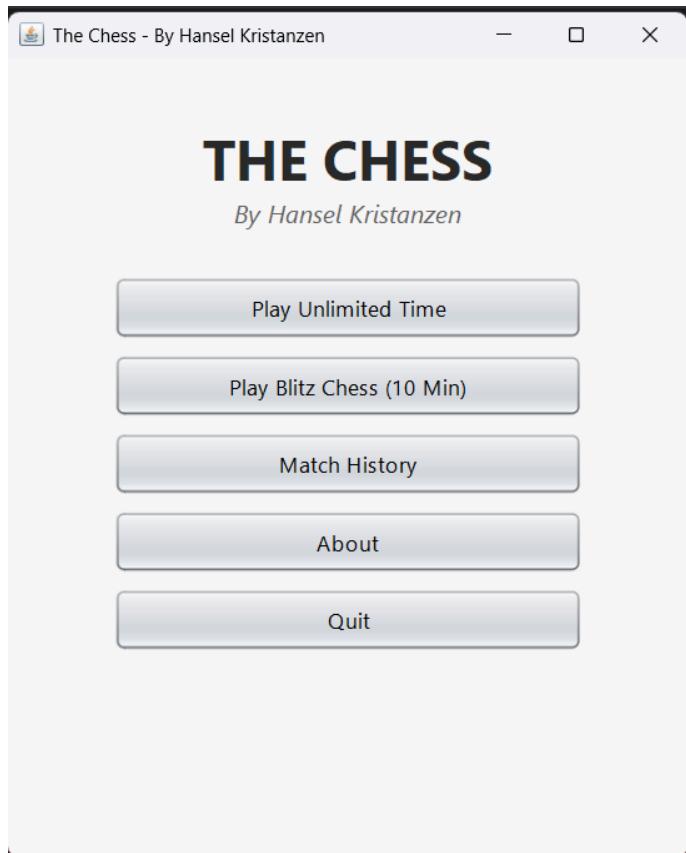
- **File I/O:** Menggunakan BufferedWriter dan BufferedReader (Java IO) untuk menyimpan data log permainan ke file chess_log.csv. Ini memastikan data tetap ada meskipun aplikasi ditutup.
- **Collections:**
 - ArrayList<String>: Digunakan untuk menyimpan riwayat langkah sementara di memori.
 - Vector: Digunakan di DefaultTableModel untuk menampilkan data di JTable.
- **Exception Handling:** Menggunakan blok try-catch untuk menangani IOException saat akses file dan IllegalArgumentException saat validasi input nama pemain.

Cara Menjalankan

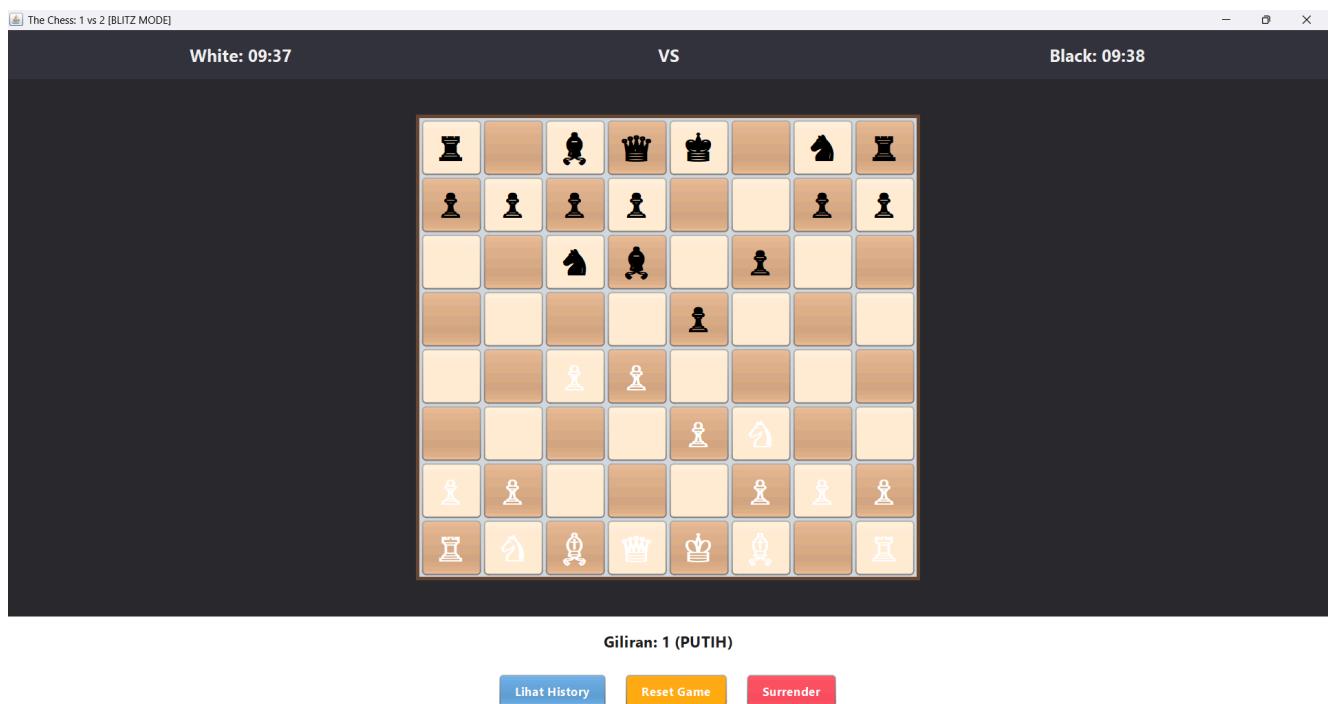
1. Pastikan Java Development Kit (JDK) sudah terinstall (Minimal JDK 8).
2. Buka IDE
3. Open folder ChessGame
4. Buka file src/chess7/main/Main.java
5. Klik tombol Run
6. Pilih menu "Play Unlimited" atau "Play Blitz" untuk memulai.
7. Masukkan nama kedua pemain
8. Klik Piece lalu klik kotak tujuan yang valid

Screenshot Aplikasi

1. Main Menu



2. Gameplay (Board)



3. Match History

Waktu	Giliran	Bidak	Dari	Ke	Keterangan
2026-01-08 00:26:52	Budi	Pawn	D2	D4	Gerak Normal
2026-01-08 00:26:54	Siska	Pawn	E7	E5	Gerak Normal
2026-01-08 00:26:57	Budi	Pawn	E2	E3	Gerak Normal
2026-01-08 00:26:59	Siska	Pawn	D7	D5	Gerak Normal
2026-01-08 00:27:02	Budi	Pawn	D4	E5	Memakan Pawn
2026-01-08 00:27:04	Siska	Queen	D8	D6	Gerak Normal
2026-01-08 00:27:06	Budi	Pawn	E5	D6	Memakan Queen
2026-01-08 00:27:08	Siska	King	E8	D7	Gerak Normal
2026-01-08 00:27:13	Budi	Queen	D1	D4	Gerak Normal
2026-01-08 00:27:16	Siska	Bishop	F8	D6	Memakan Pawn
2026-01-08 00:27:19	Budi	Knight	G1	F3	Gerak Normal
2026-01-08 00:27:25	Siska	Bishop	D6	E5	Gerak Normal
2026-01-08 00:27:28	Budi	Queen	D4	C5	Gerak Normal
2026-01-08 00:27:29	Siska	King	D7	D6	Gerak Normal
2026-01-08 00:27:30	Budi	Queen	C5	D6	CHECKMATE/WIN

Tutup

Link Demo Video

Link video presentasi: <https://youtu.be/WavXuBRbfuw>